**Experiment Title:** Write a program to manage session in PHP having login facility in any web application

# Objective:

To learn and implement a program to manage session in PHP having login facility in any web application

# Theory & Procedure :

A session is a way to store information (in variables) to be used across multiple pages.

Unlike a cookie, the information is not stored on the users computer.

**What is a PHP Session?**

When you work with an application, you open it, do some changes, and then you close it. This is much like a Session. The computer knows who you are. It knows when you start the application and when you end. But on the internet there is one problem: the web server does not know who you are or what you do, because the HTTP address doesn't maintain state.

Session variables solve this problem by storing user information to be used across multiple pages (e.g. username, favorite color, etc). By default, session variables last until the user closes the browser.

So; Session variables hold information about one single user, and are available to all pages in one application.

**Tip:** If you need a permanent storage, you may want to store the data in a [database](https://www.w3schools.com/pHp/php_mysql_intro.asp).

**Start a PHP Session**

A session is started with the session\_start() function.

Session variables are set with the PHP global variable: $\_SESSION.

Now, let's create a new page called "demo\_session1.php". In this page, we start a new PHP session and set some session variables:

Example

<?php  
// Start the session  
session\_start();  
?>  
<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
// Set session variables  
$\_SESSION["favcolor"] = "green";  
$\_SESSION["favanimal"] = "cat";  
echo "Session variables are set.";  
?>  
</body>  
</html>

**Note:** The session\_start() function must be the very first thing in your document. Before any HTML tags.

**Get PHP Session Variable Values**

Next, we create another page called "demo\_session2.php". From this page, we will access the session information we set on the first page ("demo\_session1.php").

Notice that session variables are not passed individually to each new page, instead they are retrieved from the session we open at the beginning of each page (session\_start()).

Also notice that all session variable values are stored in the global $\_SESSION variable:

**Example**

<?php  
session\_start();  
?>  
<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
// Echo session variables that were set on previous page  
echo "Favorite color is " . $\_SESSION["favcolor"] . ".<br>";  
echo "Favorite animal is " . $\_SESSION["favanimal"] . ".";  
?>  
  
</body>  
</html>

Another way to show all the session variable values for a user session is to run the following code:

Example

<?php  
session\_start();  
?>  
<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
print\_r($\_SESSION);  
?>  
  
</body>  
</html>

**How does it work? How does it know it's me?**  
Most sessions set a user-key on the user's computer that looks something like this: 765487cf34ert8dede5a562e4f3a7e12. Then, when a session is opened on another page, it scans the computer for a user-key. If there is a match, it accesses that session, if not, it starts a new session.

**Modify a PHP Session Variable**

To change a session variable, just overwrite it:

Example

<?php  
session\_start();  
?>  
<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
// to change a session variable, just overwrite it   
$\_SESSION["favcolor"] = "yellow";  
print\_r($\_SESSION);  
?>  
  
</body>  
</html>

**Destroy a PHP Session**

To remove all global session variables and destroy the session, use session\_unset() and session\_destroy():

Example

<?php  
session\_start();  
?>  
<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
// remove all session variables  
session\_unset();   
// destroy the session   
session\_destroy();   
?>  
</body>  
</html>

# Conclusion: